Scrutinizing Cross-Platform Mobile App Development Frameworks: PhoneGap and Ionic

Qurban Ali Mari[#1], Sania Bhatti[*2], Tariq Jameel Saifullah Khanzada[#3]

[#]*Computer systems engineering Department, Mehran University of Engineering & Technology) Jamshoro, Sindh, Pakistan*

[*]*Software engineering Department, Mehran University of Engineering & Technology) Jamshoro, Sindh, Pakistan*

Abstract *In present era, mobile application development is becoming more challenging with multiple platforms and their software development kits. In order to overcome the development budget and reach out to maximum users who are using multiple platforms, developers are shifting themselves towards cross-platform application development frameworks. In this research, we will scrutinize two frameworks of cross application development. First a small-scale mobile application is developed in both frameworks (PhoneGap, Ionic) then the performance of the frameworks is analysed based on seven measures. Out of which five are quantitative measures. These quantitative measures include CPU usage, memory usage, power consumption, start-up timings and lines of code (LOC's). After comparison it is found that application developed using Ionic consumes less memory, less CPU usage and less LOC's than PhoneGap. And it was also deduced that PhoneGap takes more start-up time, since it does not include dedicated user interface components.*

Keywords — *mobile development, cross platform, PhoneGap, Ionic, frameworks*

I. INTRODUCTION

Smartphones and tablets industry have been increasing steadily over the past decade. For multiple types of software applications, this smartphone industry has become a vital growth platform. Smart devices are fragmented as a technological platform. An independent software vendor must create the same application for various mobile operating systems in order to achieve the same user coverage on different smart devices because of the compatibility differences of software development kits [6] of mobile operating systems. This problem increases the importance of cross platform frameworks [1-5] for application developers.

In this research, we have scrutinized the two frameworks of cross applications development which are Ionic [2, 4] and phoneGap [3]. Two frameworks are scrutinized by a case study in which we have analyzed the major factors that will help developers and users. The aim of this research is to perform the comparative analysis and explore data related to selected frameworks for analysis of their performance. We implemented the "TicBooking" application both with PhoneGap and Ionic. We compared the implementations and found that the PhoneGap implementation has a considerable higher start-up time, package size, and memory requirement. However, all parameters were still within the acceptable range for our prototype [1]. The TicBooking application is about purchasing of tickets for variety of events. The categories of ten events include arts and culture, career and business, dancing, education and learning, film and media, food and drink, games, health, hobbies and craft, community. Two versions of the same TicBooking application are developed; one using the Ionic and other via PhoneGap frameworks.

The chief contribution of this work is the development and comparison of the app development using cross platform mobile application development technology. Then comparison of the choices of optimal constraints for mobile application development frameworks. In last developer tests and evaluates two selected frameworks and perform comparative analysis. And evaluate which framework is more suitable and resources saver among both. The results of a case study reveal that while the cross-platform technique was easy to use, the appearance and usability of the application was set at its best. In the application development, we have used ionic native plugins like camera, photolibrary, filetransfer and web services to make connection between server and mobile applications. This paper is organized in given sequence: There is introduction in section I, literature review is placed at section II. After that in section III, tools are listed. Then section IV is about implementation. Section V is about results and discussions. Section VI states the conclusions.

II. RELATED WORK

There have been past studies that mainly focused on a qualitative as well as quantitative analysis and evaluations of cross-platform development, analyzing the available support to the frameworks,

programming languages, and development environment. The first comparative evaluation of usability of cross platform applications using PhoneGap was presented in literature by Mesfin, G. et al. [1]. They developed one application for Android, Blackberry and Windows phone with PhoneGap and conducted the usability testing. Authors in study [6] claimed after evaluating the usability of cross-platform smartphone applications that because of the JavaScript runtime needed to operate in cross-development approaches, the cross-platform applications is more CPU intensive and cost more memory compared to their natively developed counterpart to functioning properly. The JavaScript approach also shows to have the slowest launch time when measuring the diff erent developed applications, however, when the application is fully loaded, navigation and response times are generally like native response times. In study [7] authors compared the Ionic, PhoneGap, NativeScript and Native app based on computer and mobile platform support, document support and development support. They supported the PhoneGap due to the widespread range of supported platforms. In another study [8] authors compared the performance of build, rendering and user interface response time of Cordova, Native app, Titanium, Xamarin by developing a set of mobile apps with similar features. They concluded the significant differences and trade-offs of compared approaches. Authors in their work presented in [9] compared the ReactNative, Ionic and Fuse frameworks. They presented the supported platforms, app performance and testability. They concluded without suggesting the best framework among these three choices. The researchers in a study [10] suggests that PhoneGap is reasonable alternative for a mobile application after developing and analyzing an application with three different techniques. The study suggested that instead of developing the application discretely for each target platform, a cross-platform solution can be a feasible option. A detailed discussion and categorization of cross platform development approaches in presented by authors in study [11]. Recently, Rieger and Majchrzak [12] have proposed an evaluation framework to access cross platform app development approaches. The framework has compared numerous app development approaches by considering weight profiles. Biørn-Hansen and Ghinea [13] compared the Hybrid-based Ionic app is written in a standard JavaScript and the interpreted react native app written in new JavaScript. They experimented with quantitative analysis based on file system access performance.

III. TOOLS

A. *Visual Studio*

For coding purpose there are multiple tools to use but Visual studio's huge libraries, built in support, packages and user-friendly environment provide more easiness to developer. More ever It is an IDE for developing the cross-platform software for multiple OS as Android, Windows and iOS. With more advanced build and debugging support.

B. *NODE.JS*

In this regard Node.js is designed to make scalable network software for cross app development. Event-driven, non-blocking I/O model features makes Node.js efficient and light weighted and perfect for data-intensive real-time applications which run over distributed devices. Node.js is a cross-platform, open source runtime environment for creating server-side and networking applications.

C. *CORDOVA*

In Cordova is a collection of open source device APIs which allows the developed applications to use native device functions across the several device platforms. Using the Cordova's APIs, and plug-ins of cross-platform, developers can build and code applications with help of HTML5, CSS3 and Java Web based languages.

IV. IMPLEMENTATION

A. *Ionic Framework services and plugins*

In TicBooking application, we have used ionic native plugins including camera, photolibrary, filetransfer and webservices to make connection between server and mobile app. In this app Camera-plugin was used to get image of ticket buyer from device gallery or choosing images from the system's library to select image so that it could be checked at time of arrival. In this app ionic file transfer plugin was used to upload ticket from user's device. After selecting the image this plugin will help in uploading image. In our application client will login and will be connected to the internet through server and after login, form of details will be sent to the server. And web also supports two-way communication.

B. *PhoneGap Framework services and plugins*

In TicBooking app we have used jquery functions like camera,photolibrary, filetransfer and webservices to make connection between server and mobile apps. Camera is a powerful, configurable, responsive, mobile-friendly jQuery slideshow/slider/carousel plugin that comes with fancy slice transition effects between slides. In this app Camera-plugin was used to get image of ticket buyer from device gallery to select image so that it could be checked at time of arrival. File transfer plugin was used to upload ticket from user's device. After selecting the image this plugin will help in uploading image.

C. *Working of TicBooking app*

The layout of the TicBooking application is shown in Fig. 1. Application activities includes: Login, Event ticket categories and Post event with payment details.

The side by side interfaces of Login Pages of TicBooking application developed in Ionic and Phonegap are shown in Fig. 2(a) and Fig. 2(b) respectively. Both pages need Email address and Password from user. After login user will enter to the Category page activity.

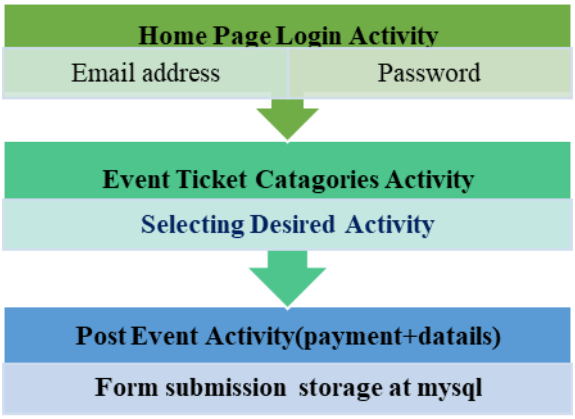


Fig 1: Layout of the TicBooking application

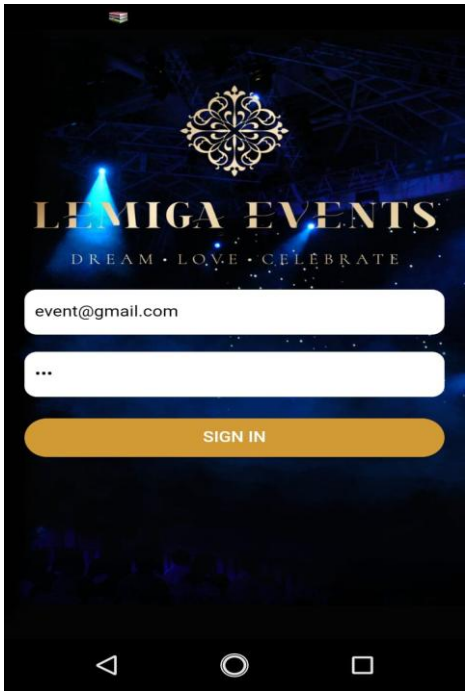
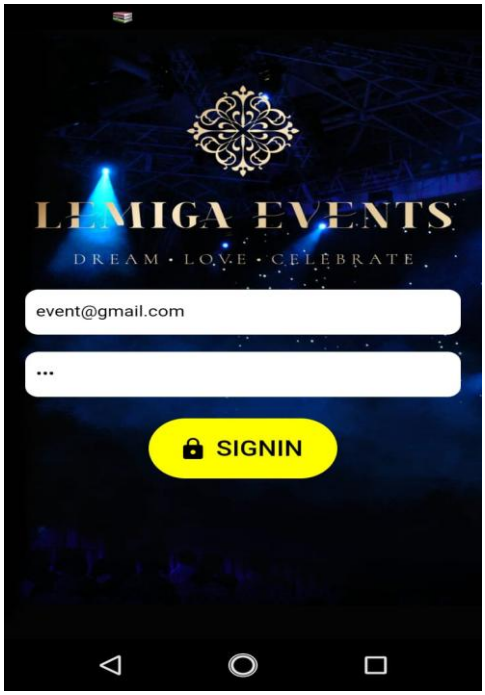


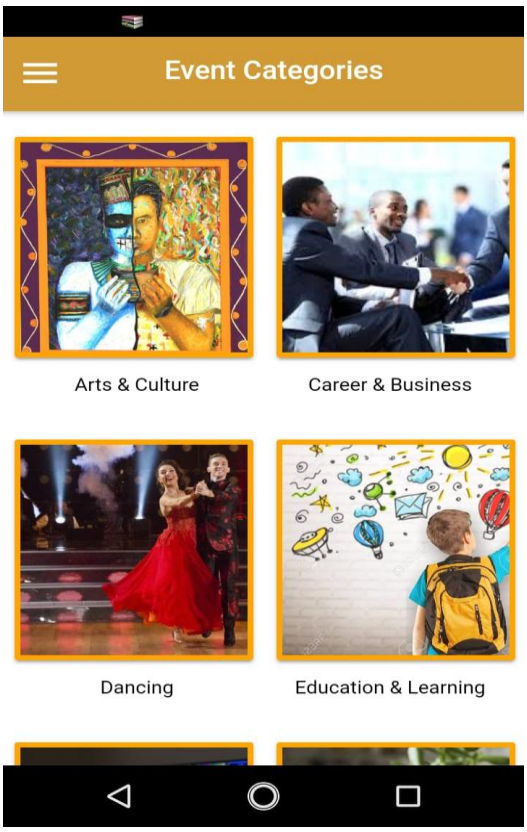
Fig 2: (a) Ionic Login Page



(b) PhoneGap Login Page

In Event categories activity page approximately 10 categories are listed. The categories include arts and culture, career and business, dancing, education and learning, film and media, food and drink, games, health, hobbies and craft, community. The side by side interfaces of event categories of TicBooking application developed in Ionic and Phonegap are shown in Fig. 3(a) and Fig. 3(b) respectively.

The side by side interfaces of event detail page of TicBooking application developed in Ionic and Phonegap are shown in figure 4(a) and figure 4(b) respectively. This demonstrates the view of ticket that user had selected where users image and all details will be saved.



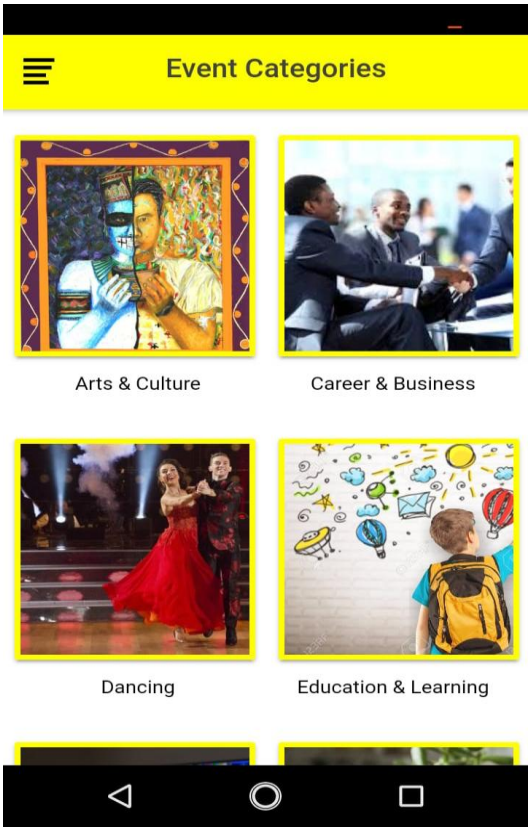
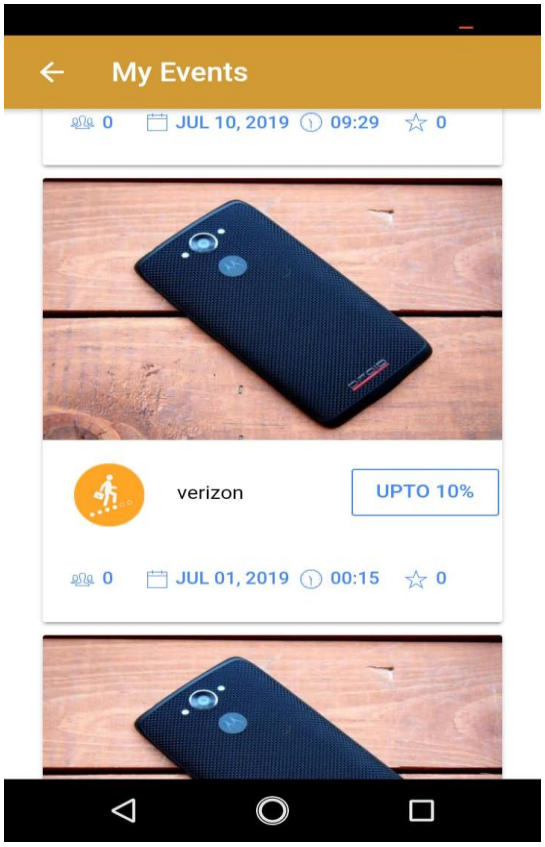


Fig 3: (a) Ionic Categories Page **(b) PhoneGap Login Page**



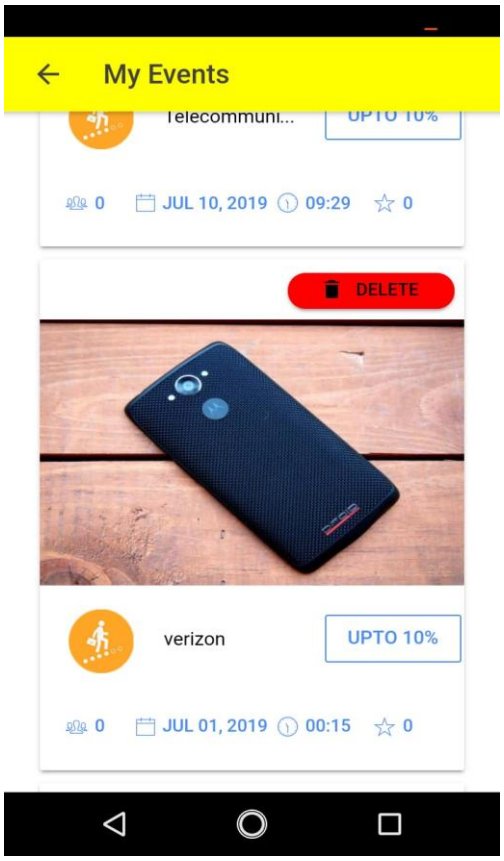


Fig 4: (a) Ionic Categories Page **(b) PhoneGap Login Page**

V. RESULTS & DISCUSSION

The experiments are conducted on Samsung S8 running Android 7.1.1. The CPU usage and LOC are measured from the HAIER laptop CORE i5 running Windows 8. For comparing the performance of the two frameworks, five quantitative measures are evaluated, and the results shown and discussed in this section. These quantitative measures are CPU usage, memory usage, power consumption, startup timings and LOC's.

A. Start-up Time measures

The start-up timings of four pages of TicBooking application developed using Ionic and PhoneGap frameworks are depicted in Fig. 5. Ionic application takes 5 seconds and PhoneGap application takes upto 8 seconds of start-up time. After comparing the start-tup timings of four pages i.e. Home page, Category Page, MyEvent page and Event details page it has been deduced that pages of TicBooking application developed in PhoneGap takes more time to start the pages than the TicBooking application developed in Ionic.

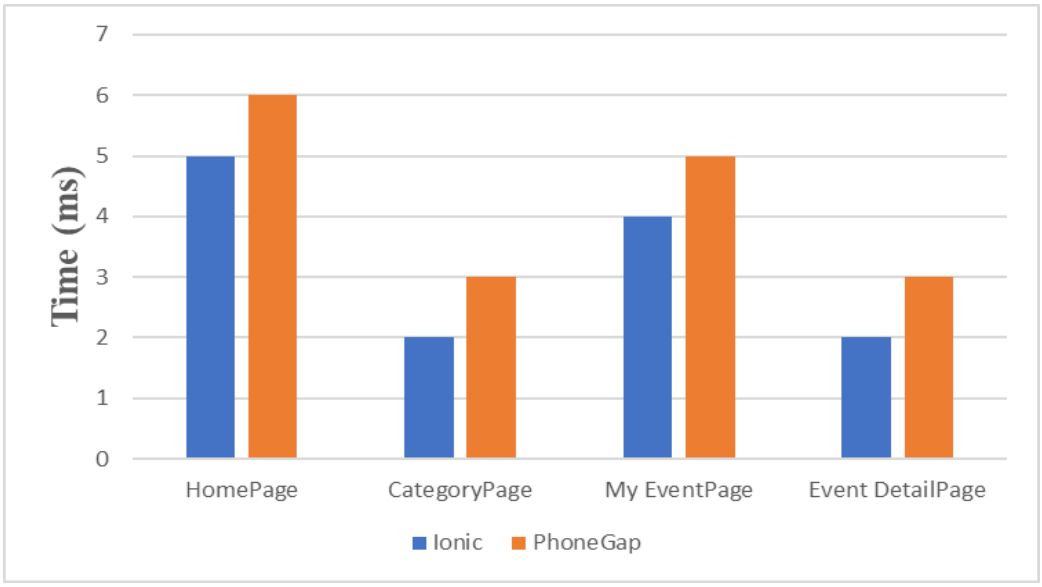


Fig 5: Comparison of Start-up timing of TicBooking app pages developed in Ionic and PhoneGap frameworks

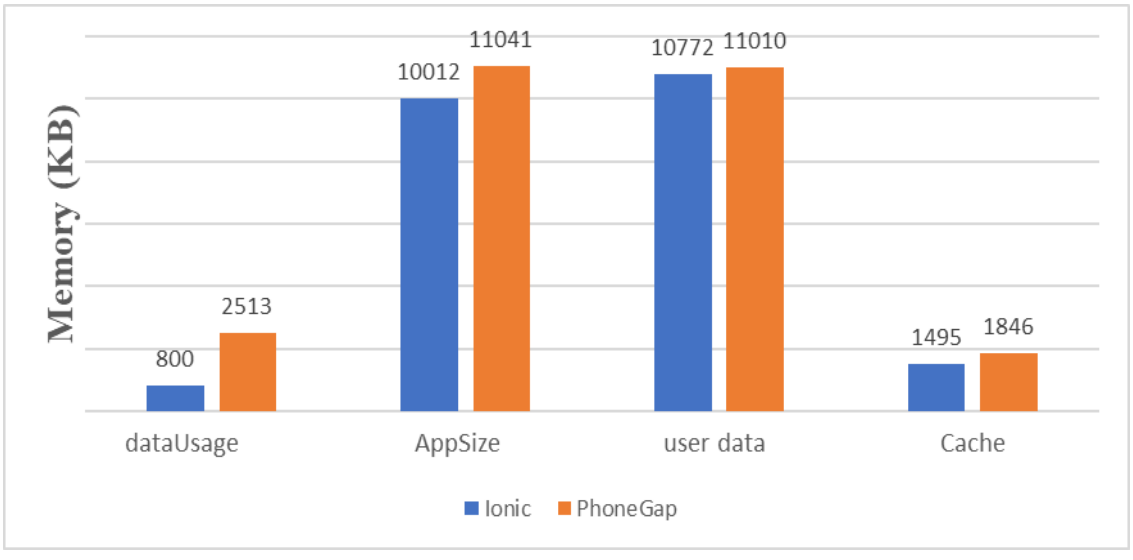


Fig 6: Comparison of Memory usage of TicBooking application developed in Ionic and PhoneGap frameworks

B. Memory Usage measures

The memory usage of TicBooking application developed using Ionic and PhoneGap frameworks are depicted in Fig. 6. Memory usage considers data usage, App size, user data and cache in KBs. After comparing the memory usage of four measures it is evident that pages of TicBooking application developed in PhoneGap occupies slightly more memory than the TicBooking application developed in Ionic. TicBooking application developed in Ionic data usage is 800kbs, app size needs 10012 kBs, user data needs 10772kbs and cache occupies 1495 kBs of memory. TicBooking application developed in PhoneGap data usage is 2513 kBs, app size is 11041 kBs, user data needs 11010kbs. Cache occupies 1846 kBs of memory.

C. CPU Usage measures

CPU usage can be monitored to see how much of the processor's capacity is in use. The CPU usage of TicBooking application developed using Ionic and PhoneGap frameworks is depicted in Fig. 7. TicBooking application developed via Ionic framework CPU usage is 13% which is good and optimized as compared to TicBooking application developed in Phonegap.

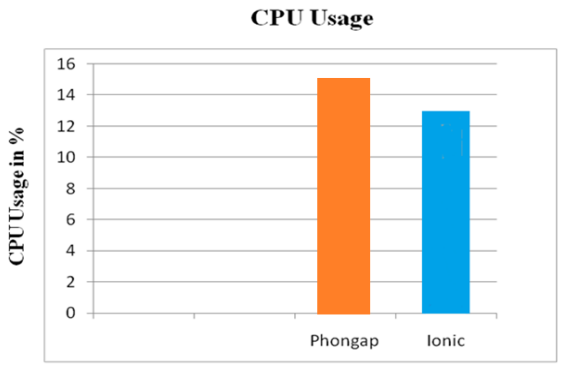


Fig. 7 Comparison of CPU usage of TicBooking application developed in Ionic and PhoneGap frameworks.

D. Power Consumption

Power consumption always affected by execution time and the execution time is affected by the CPU usage. The more execution time consumes more energy. TicBooking application developed via Ionic framework consumes smaller power than TicBooking application developed in Phonegap [14-16].

The performance of both the frameworks have been evaluated via quantitative and qualitative measures which are summarized in table I. It is evident from the table that the application developed using ionic framework performs better than the application developed using PhoneGap in terms of less memory, CPU usage, smaller start-up time, power consumption and reduced number of LOCs.

Table I. Quantitative and qualitative measures for performance evaluation of Ionic and PhoneGap frameworks

Quantitative Measures	**Ionic**	**PhoneGap**
Memory usage	12.5 MB	16 MB
CPU usage	13% CPU	15% CPU
Start-up Time	5 seconds	8 seconds
Power consumption [15]	15 Joule	17 Joule
Lines of Code	1250 lines	1819 lines
Qualitative Measures	**Ionic**	**PhoneGap**
GUI	Its own GUI	UI from outside (not its own)
Usage	Best for rapid & low cost[1-10]	Best for light apps [1-10]

VI. CONCLUSIONS

The choice of the proper framework for cross-platform app development completely depends on the developer. Both Ionic and PhoneGap have their own advantages and disadvantages, as well as a unique set of features. PhoneGap is perfect for creating light applications that don't rely on the device's native features. Ionic allows for the rapid and application development at zero costs. The major difference between Ionic and PhoneGap is pricing. After analyzing the quantitative and qualitative measures it was deduced that the performance of ionic is better than phoneGap.

The study has analyzed the two frameworks by developing a small-scale thin client mobile application. The study can be extended by developing additional intricate mobile application employing the diverse advance features.

ACKNOWLEDGMENT

The authors are thankful to MUET for conducting and supporting this research work.

REFERENCES

[1] G. Mesfin, G. Ghinea, D. Midekso, T.M. Grønli, "Evaluating Usability of Cross-Platform Smartphone Applications", in Proc. 11th International Conference on Mobile Web Information Systems (MobiWIS 2014), 2014 LNCS 8640, pp. 248-60.

[2] Ioinic website. [Online]. Available: https://ionicframework.com/getting-started/.

[3] PhoneGap website. [Online]. Available: http://phonegap.com/.

[4] Kylie "Building a Mobile App: Ionic Vs React Native Vs Native". [Online]. Available: https://simpleweb.co.uk/building-a-mobile-app-ionic-vs-react-vs-native/.

[5] V.C. Kulloli, A. Pohare, S. Raskar, T. Bhattacharyya, S. Bhure, "Cross Platform Mobile Application Development,"

International Journal of Computer Trends and Technology (IJCTT), vol., no. 5, pp. 1095-1100 May 2013.

[6] M. Willocx, J. Vossaert, and V. Naessens, "Comparing Performance Parameters of Mobile App Development Strategies," in Proc. of the International Conference on Mobile Software Engineering and Systems, 2016, pp. 38–47.

[7] T. Vil ˇcek, T. Jakopec, Comparative analysis of tools for development of native and hybrid mobile applications. In: International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1516–1521.

[8] X. Jia, A. Ebone, Y. Tan, "A performance evaluation of cross-platform mobile application development approaches," In: Proc. of the 5th International Conference on Mobile Software Engineering and Systems, 2018 ACM, New York, NY, USA, pp. 92–93.

[9] T. Majchrzak, and G. Tor-Morten "Comprehensive analysis of innovative cross-platform app development frameworks," In Proc. of the 50th Hawaii International Conference on System Sciences, 2017.

[10] A. Ville, S. Hyrynsalmi, and O. Nevalainen. "An evaluation framework for cross- platform mobile app development tools: A case analysis of adobe phonegap framework," In Proc. of the 17th International Conference on Computer Systems and Technologies, 2016.

[11] A. Biørn-Hansen, G. Tor-Morten and G. Gheorghita, "A Survey and Taxonomy of Core Concepts and Research Challenges in Cross-Platform Mobile Development," ACM Computing Surveys (CSUR), vol. 51, no.5, Article 2018, 108.

[12] C. Rieger, and T. A. Majchrzak, "Towards the definitive evaluation framework for cross-platform app development approaches," Journal of Systems and Software, vol. 153, pp. 175-199, 2019.

[13] A. Biørn-Hansen, G. Ghinea, "Bridging the gap: investigating device-feature exposure in cross-platform development," In: Proc. of the 51st Hawaii International Conference on System Sciences, 2018 ScholarSpace, pp. 5717–5724.

[14] M. Ciman, and G. Ombretta, "An empirical analysis of energy consumption of cross-platform frameworks for mobile development," Pervasive and Mobile Computing, vol. 39, pp. 214-230, 2017.

[15] J. Berrocal, et al. "Early analysis of resource consumption patterns in mobile applications," Pervasive and Mobile Computing, vol. 35, pp. 32-50, 2017.

[16] A. Merlo, M. Migliardi, and P. Fontanelli, "Measuring and estimating power consumption in Android to support energy-based intrusion detection," J. Comput. Secur., vol. 23, no. 5, pp. 611–637, Aug. 2015.